BBM 418 - Computer Vision Laboratory

HACETTEPE UNIVERSITY

Department of Computer Engineering

Programming Assignment-3
Spring 2024-2025
Dr. Nazli Ikizler Cinbis

Due Date: 23:59 on Friday, May 23rd, 2025

Object Detection and Counting from Drone-based Images

In this assignment, you will become familiar with object detection using a pre-trained YOLO (You Only Look Once) model. Your task is to count the number of cars in parking lot images captured by a drone. You will use YOLO to detect car objects, determine the number of cars in each image, and compare your results with the ground-truth annotations.

You Only Look Once (YOLO)

YOLO is a fast object detection algorithm that predicts bounding boxes and class labels for objects in an image in a single pass through a neural network. It divides the image into a grid, and for each grid cell, it directly predicts the object class and its location. Unlike older methods that first generate region proposals and then classify them separately, YOLO treats detection as a single regression problem, making it extremely fast and efficient for real-time applications.

YOLO-v8

YOLOv8 [1] is the latest version of the YOLO object detection model, offering improved speed, accuracy, and flexibility. It keeps the core idea of detecting objects in a single pass but replaces the older anchor-based system with an anchor-free approach, making the model simpler and more effective, especially for small or irregular objects.

It also expands beyond detection, supporting segmentation, classification, and pose estimation. With a redesigned architecture and easy export options (like ONNX and TensorRT), YOLOv8 is more powerful and easier to deploy than earlier versions.

You can find brief model architecture for YOLO-v8n model on YOLOv8n_model.txt file.

Dataset

The Car dataset contains about 1400 drone-based parking lot images. In this dataset, you will focus on only cars, not other vehicles. This dataset is a small portion of the CARPK dataset [2, 3]. For each image there is a text file that contains the annotation for cars in the image. ***All labeled bounding boxes have been well recorded with the top-left points and the bottom-right points.*** The dataset is separated into 1000 images for training, 200 images for validation, and 200 images for testing, and you can find the splitting files in the ImageSets folder. Figure 1 includes some examples of parking lot images and annotation of cars with YOLO-v8. The dataset folder structure is given below.

cars_dataset

- Images
- Annotations
- ImageSets

Download Link: cars_dataset

Figure 1: Examples for dataset and their annotations with YOLO-v8.

Implementation Details

Code Part

1. Prepare your dataset before starting fine-tuning on YOLO-v8. [**10 pts**]
   - YOLOv8n (nano) is the fastest and lightest model in the YOLOv8 family and is used for real-time detection and Edge deployment (e.g., Jetson Nano, Raspberry Pi) [1].
2. In this assignment, you are required to train your model under four different settings:
   - **Freeze the first 5 blocks:** This includes the stem (initial convolution) and the first two backbone (C2f) blocks. [**5 pts**]
   - **Freeze the first 10 blocks:** This includes the entire backbone along with the SPPF (Spatial Pyramid Pooling - Fast) bottleneck layer. [**5 pts**]
   - **Freeze the first 21 blocks:** All layers are frozen except the Detection head. [**5 pts**]
   - **Train the entire model:** No layers are frozen; the full network is trained. [**5 pts**]
3. For the given models, use different hyper-parameters (optimizer, learning rate, batch size), their effects, with the loss plots and visualized bounding box predictions.
4. After training, you will compare the number of cars predicted in each image with the ground-truth (annotations). You will measure exact match accuracy and MSE (Mean Squared Error).
   - The exact match accuracy metric calculates the percentage of images where the predicted count exactly equals the ground truth count. [**5 pts**]

$$\text{Exact Match Accuracy} = \frac{\text{Number of exact matches}}{\text{Total number of images}} \times 100$$

- The mean squared error measures the average squared difference between the predicted and actual number of cars. **[5 pts]**
  Where:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(Predicted_i - GroundTruth_i)^2$$

  - N is the number of evaluated images
  - $Predicted_i$ is the predicted number of cars for image i
  - $GroundTruth_i$ is the true number of cars in image i

What should you write in the report?

1. Give explanations for each step. **[10 pts]**
2. Give your models' loss plots for both the training and validation sets during training. **[5 pts]**
3. Evaluate the results of **learning hyperparameters** (choice of optimizer, learning rate, batch size, etc.), their effects on the performance, together with associated loss plots. Discuss the results in detail. **[10 pts]**
4. Explore the results of **different hyperparameters specific to the YOLO architecture**, their effects on the performance, together with associated loss plots. Discuss the results in detail. **[10 pts]**
5. Visualize the bounding box predictions, give examples of correct detections and missed regions. **[5 pts]**
6. Compare and analyze the results of the best performing versions of object detectors that are acquired by the above 4 training settings over the test set. Evaluate their performance in terms of Exact match accuracy, MSE, Precision and Recall. Please state your observations clearly and precisely. What can be done to improve the performance?**[10 pts]**
7. What other types of methods can be used to solve similar object counting problems? Explain alternative methods briefly. **[10 pts]**

What to Hand In

Your submission format will be:

- b<studentNumber>.ipynb (iPython notebook)(Report on the iPython file)
- b<studentNumber>.py (Convert your ipynb file to py file)

Archive this folder as **b<studentNumber>.zip** and Google Classroom.

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out abstractly. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else. You may not share or receive concrete solutions such as code, pseudocode, or detailed proof steps, nor may you copy, adapt, or refine text or code produced by LLM tools (e.g., ChatGPT, Gemini, Copilot) or any online source.

References

[1] `https://docs.ultralytics.com/models/yolov8/`

[2] `https://lafi.github.io/LPN/`

[3] Hsieh, M. R., Lin, Y. L., & Hsu, W. H. (2017). Drone-based object counting by spatially regularized regional proposal network. In Proceedings of the IEEE international conference on computer vision (pp. 4145-4153).